

Contenedores LXC

Virtualización ligera en Linux

 José Domingo Muñoz

 IES Gonzalo Nazareno · Dos Hermanas

 IV · Infraestructura Virtual

IV · CONTENEDORES LXC

01

Introducción a LXC

Concepto, instalación y ciclo de vida

¿Qué es LXC?

*"Linux Containers (LXC) es una tecnología de **virtualización ligera**: sobre el núcleo del sistema operativo se ejecuta una capa de virtualización que permite que existan **múltiples instancias aisladas de espacios de usuario**. A estas instancias las llamamos **contenedores**."*

CARACTERÍSTICAS PRINCIPALES

- Pertenece a los **contenedores de sistema**: su gestión y ciclo de vida son similares a los de una máquina virtual tradicional
- Se accede a ellos por SSH, se instalan servicios, se actualizan...
- Lo desarrolla la comunidad del proyecto **Linux Containers**; página oficial: linuxcontainers.org

Instalación

```
apt install lxc
```

Junto a `lxc` se instalan sus paquetes **recomendados**; dos de ellos hacen falta en el curso:

- **apparmor**: confina los contenedores con el perfil que genera LXC (`lxc.apparmor.profile = generated`)
- **lxcfs**: hace que `free` o `/proc/cpuinfo` muestren, dentro del contenedor, los recursos limitados

⚠ Sin `apparmor`, el contenedor no arranca: `lxc-start` da `Failed to receive the container state` y, en primer plano (`lxc-start -n contenedor1 -F`), `Cannot use generated profile: apparmor_parser not available`. Se soluciona con `apt install apparmor`.

Tecnologías del kernel que lo hacen posible

GRUPOS DE CONTROL (*CGROUPS*)

- En Debian se usa **cgroups v2**
- **Limitan el uso de recursos** de un proceso y sus hijos: memoria, CPU, E/S de disco...

ESPACIOS DE NOMBRES (*NAMESPACES*)

- Proporcionan una **vista aislada** del sistema a cada contenedor
- Aíslan: **interfaces de red**, procesos, usuarios, sistema de ficheros...

i A diferencia de las MV, todos los contenedores **comparten el kernel del host**. No hay hipervisor: el aislamiento lo provee el propio kernel.

Contenedores privilegiados y no privilegiados

PRIVILEGIADOS

- Se crean y se ejecutan como `root`
- Configuración en `/var/lib/lxc/`
- Son los que **usamos en el curso**

NO PRIVILEGIADOS

- Se crean y se ejecutan con un **usuario normal**
- Más seguros: el `root` del contenedor no es el `root` del host

 Todos los comandos `lxc-*` de esta presentación se ejecutan como `root`.

Creación de contenedores — plantilla `download`

```
lxc-create -n contenedor1 -t download
```

`-n` : nombre del contenedor · `-t download` : descarga la imagen del servidor de imágenes

1. Muestra la **lista de distribuciones** disponibles
2. Pide la **distribución**, la **versión** y la **arquitectura**
3. La primera vez descarga el sistema de ficheros comprimido (**plantilla**) y lo guarda en caché en `/var/cache/lxc`
4. Se descomprime la plantilla y se crea el sistema de ficheros del contenedor en `/var/lib/lxc/contenedor1/rootfs/`
5. Los siguientes contenedores con la misma distribución, versión y arquitectura se crean **más rápido**: no se vuelve a descargar

Creación no interactiva

Indicamos la distribución, la versión y la arquitectura después de `--`:

```
lxc-create -n contenedor2 -t download -- -d debian -r trixie -a amd64
```

OPCIÓN	SIGNIFICADO
<code>-d</code>	Distribución (<i>distribution</i>)
<code>-r</code>	Versión (<i>release</i>)
<code>-a</code>	Arquitectura (<i>arch</i>)

 Las imágenes disponibles están en el servidor `images.linuxcontainers.org`.

Gestión básica de contenedores

```
lxc-ls [-f]                # Listar contenedores (-f: con detalles)

lxc-start  contenedor1      # Arrancar
lxc-stop   contenedor1      # Parar
lxc-stop -r contenedor1     # Reiniciar

lxc-attach contenedor1      # Consola interactiva
lxc-attach contenedor1 -- ls -al  # Ejecutar un comando (contenedor arrancado)
lxc-execute contenedor1 -- ls -al  # Ejecutar un comando (contenedor parado)

lxc-destroy [-f] contenedor1 # Eliminar contenedor
```

```
$ lxc-ls -f
NAME          STATE   AUTOSTART GROUPS IPV4      IPV6 UNPRIVILEGED
contenedor1   RUNNING 0        -      10.0.3.180 -      false
```

Información de un contenedor

```
lxc-info contenedor1
```

OPCIÓN	MUESTRA
<code>-i</code>	Solo la dirección IP
<code>-S</code>	Estadísticas de uso: CPU, E/S de disco, memoria y datos enviados y recibidos por la red
<code>-s</code>	El estado del contenedor

 La columna `UNPRIVILEGED` de `lxc-ls -f` indica si el contenedor es no privilegiado: en nuestro caso, `false`.

02

Configuración de LXC

Fichero de configuración y límite de recursos

Ficheros de configuración

FICHERO	ALCANCE
<code>/etc/lxc/default.conf</code>	Plantilla de configuración: la usan los contenedores nuevos
<code>/var/lib/lxc/<nombre>/config</code>	Configuración de cada contenedor (se genera a partir de la plantilla)

 Si cambiamos `/etc/lxc/default.conf`, los **nuevos contenedores** tendrán la configuración indicada; los que ya existen no cambian.

Fichero de configuración de contenedor1

/var/lib/lxc/contenedor1/config:

```
# Distribution configuration
lxc.include = /usr/share/lxc/config/common.conf
lxc.arch = linux64

# Container specific configuration
lxc.apparmor.profile = generated
lxc.apparmor.allow_nesting = 1
lxc.rootfs.path = dir:/var/lib/lxc/contenedor1/rootfs
lxc.uts.name = contenedor1

# Network configuration
lxc.net.0.type = veth
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up
```

Parámetros importantes

PARÁMETRO	SIGNIFICADO
<code>lxc.rootfs.path</code>	Directorio con el sistema de ficheros del contenedor (cada contenedor tiene uno completo)
<code>lxc.uts.name</code>	Nombre de host (<i>hostname</i>) del contenedor
<code>lxc.net.0.*</code>	Configuración de la interfaz de red : conectada al Linux Bridge <code>lxcbr0</code> (<code>lxc.net.0.link</code>)
<code>lxc.start.auto = 1</code>	El contenedor se inicia automáticamente al iniciar el host

⚠ Para que una modificación tenga efecto **hay que reiniciar el contenedor**.

i Todos los parámetros: linuxcontainers.org/lxc/manpages/man5/lxc.container.conf.5.html

Limitar los recursos — *cgroups*

Por defecto, un contenedor puede usar **toda la CPU, la RAM y el disco** del host. Los **grupos de control** (*cgroups*) permiten limitarlos.

Limitamos la memoria a **512 MB** y el uso a **una CPU** (la CPU 0) en `/var/lib/lxc/contenedor1/config`:

```
lxc.cgroup2.memory.max = 512M  
lxc.cgroup2.cpuset.cpus = 0
```

Reiniciamos el contenedor:

```
lxc-stop contenedor1  
lxc-start contenedor1
```

Limitar los recursos — comprobación

```
$ lxc-attach contenedor1 -- free -h
```

	total	used	free	shared	buff/cache	available
Mem:	512Mi	6.0Mi	505Mi	0.0Ki	0.0Ki	505Mi

```
$ lxc-attach contenedor1 -- cat /proc/cpuinfo
```

processor	: 0
...	

- **Memoria:** `free -h` muestra un total de **512Mi**, no la memoria del host
- **CPU:** en `/proc/cpuinfo` aparece **un solo procesador**

i Dentro del contenedor, `free` y `/proc/cpuinfo` muestran los límites gracias a **LXCFS** (paquete `lxcfs`, recomendado al instalar `lxc`). Sin él, verían los recursos del host aunque el límite se aplique.

IV · CONTENEDORES LXC

03

Redes en LXC

Bridges, red por defecto y varias interfaces

Red por defecto — `lxcbr0`

LXC ofrece varios mecanismos de red; en el curso usamos conexiones de tipo **bridge**.

Por defecto el contenedor está conectado (`lxc.net.0.link = lxcbr0`) al bridge `lxcbr0`, que crea el servicio `lxc-net`:

- Red de tipo **NAT**, con direccionamiento `10.0.3.0/24`
- El host ofrece los servicios de **DHCP** y **DNS** (`10.0.3.1`)
- Los contenedores obtienen IP automáticamente y tienen salida a Internet

```
$ lxc-ls -f
NAME          STATE   AUTOSTART GROUPS IPV4      IPV6 UNPRIVILEGED
contenedor1   RUNNING 1        -      10.0.3.180 -      false
```

Red por defecto — comprobaciones en el contenedor

```
root@contenedor1:~# ip a
2: eth0@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP ...
    inet 10.0.3.180/24 brd 10.0.3.255 scope global dynamic eth0
root@contenedor1:~# ip r
default via 10.0.3.1 dev eth0
10.0.3.0/24 dev eth0 proto kernel scope link src 10.0.3.180
root@contenedor1:~# cat /etc/resolv.conf
nameserver 10.0.3.1
root@contenedor1:~# ping www.josedomingo.org
64 bytes from ns330309.ip-37-187-119.eu (37.187.119.60): icmp_seq=1 ttl=49 time=28.7 ms
```

1. IP **10.0.3.180**, obtenida por DHCP
2. Puerta de enlace **10.0.3.1**: la máquina física
3. La máquina física es también el **servidor DNS**
4. Hay **resolución** y **acceso al exterior** (instalando antes **iputils-ping**)

Conectar a un bridge existente

Podemos conectar el contenedor a cualquier bridge, **creado por nosotros o por libvirt**. Por ejemplo, a la red `default` de libvirt (bridge `virbr0`):

1. Paramos el contenedor
2. Cambiamos en su configuración: `lxc.net.0.link = virbr0`
3. Lo iniciamos y comprobamos:

```
$ lxc-ls -f
NAME          STATE    AUTOSTART GROUPS IPV4          IPV6 UNPRIVILEGED
contenedor1  RUNNING 1        -      192.168.122.228 -      false
```

i Si el cambio se hace en `/etc/lxc/default.conf`, **todos los contenedores que creemos a partir de entonces** se conectarán a la red `default`.

Conectar un contenedor a varias redes

Cada interfaz se declara con `lxc.net.X`: `0` la primera, `1` la segunda...

```
lxc.net.0.type = veth
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up

lxc.net.1.type = veth
lxc.net.1.link = virbr0
lxc.net.1.flags = up
```

Paramos y arrancamos el contenedor (`lxc-stop` y `lxc-start`).

⚠ La nueva interfaz hay que configurarla **dentro del contenedor**, según el mecanismo de configuración de red que tenga instalado (*ifupdown*, *systemd-networkd*, *netplan*...).

Instalación de netplan en lxc

En los contenedores **Debian** la red la gestiona `systemd-networkd`; podemos configurarla con **netplan** (en los contenedores **Ubuntu** ya viene instalado):

```
apt install netplan.io
# Creamos /etc/netplan/10-lxc.yaml; debe tener permisos restrictivos:
chmod 600 /etc/netplan/10-lxc.yaml
```

```
# /etc/netplan/10-lxc.yaml
network:
  version: 2
  ethernets:
    eth1:
      addresses: [10.0.0.10/24]
```

Se aplica con `netplan apply`.

netplan apply en el contenedor

netplan apply da error porque en el contenedor **no está udev**. Se puede simular que está:

```
mkdir -p /usr/local/bin
printf '#!/bin/sh\nexit 0\n' > /usr/local/bin/udevadm
chmod +x /usr/local/bin/udevadm
```

Después, netplan apply ya funciona.

 /usr/local/bin va antes que /usr/bin en el PATH: netplan encuentra este udevadm, que no hace nada y termina sin error.

IV · CONTENEDORES LXC

04

Almacenamiento en LXC

Montar directorios del host

Montar un directorio del host

Montamos `/opt/contenedor1` del host en `/srv/www` del contenedor:

1. Crear el punto de montaje **dentro del contenedor**:

```
root@contenedor1:~# mkdir /srv/www
```

2. Añadir la entrada en `/var/lib/lxc/contenedor1/config` :

```
lxc.mount.entry=/opt/contenedor1 srv/www none bind 0 0
```

⚠ La ruta de destino es **relativa** (sin la barra inicial) al sistema de ficheros del contenedor:

```
/var/lib/lxc/contenedor1/rootfs/.
```

Montar un directorio del host — comprobación

3. Reiniciar el contenedor y comprobar el montaje:

```
$ lxc-stop contenedor1
$ lxc-start contenedor1
$ lxc-attach contenedor1
root@contenedor1:~# cd /srv/www
root@contenedor1:/srv/www# ls
index.html
```

 El fichero `index.html` está en `/opt/contenedor1` del host: los cambios que se hagan en el host se ven en el contenedor y al revés.

IV · CONTENEDORES LXC

05

LXD e Incus

Herramientas de gestión de contenedores y MV

LXD

" **LXD** (*Linux Container Daemon*) es una herramienta de **gestión** de contenedores y máquinas virtuales para Linux, desarrollada por Canonical.

CARACTERÍSTICAS

- Ofrece una **REST API** accesible desde línea de comandos o herramientas de terceros
- Gestiona dos tipos de **instancias**:
 - **Contenedores** — usando LXC internamente
 - **Máquinas virtuales** — usando QEMU/KVM internamente
- Sus comandos (`lxc list`, `lxc launch ...`) pueden causar confusión con los de LXC (`lxc-ls`, `lxc-start ...`)

Incus

*"**Incus** es un fork de LXD que también gestiona contenedores y máquinas virtuales. Forma parte del proyecto **Linux Containers**.*

LXD

- Desarrollado por **Canonical**
- Integrado en el ecosistema Ubuntu
- Proyecto original

INCUS

- Fork de la comunidad
- Proyecto **Linux Containers**
(linuxcontainers.org/incus/)
- Alternativa independiente de Canonical

 Ambos: una única herramienta para gestionar **contenedores de sistema y MV**.

IV · CONTENEDORES LXC

¡Gracias!

Contenedores LXC

 José Domingo Muñoz

 IES Gonzalo Nazareno · Dos Hermanas

 IV